

# On Probability Density Estimation by using NNs

A. Juste

- Estimating a probability density function in n dimensions is a **necessity** which one may easily encounter in Physics and other fields.
- Some applications of PDE in HEP:
  - confidence level calculations
  - measurement of physical constants via **parametric** (maximum likelihood) or non-parametric (Bayesian method) "fitting"
  - discrimination tasks: construction of optimal discriminant, event selection and cut optimization,...
- Two approaches to PDE from the empirical pdf  $\sum_{e=1}^N \delta(\vec{x} - \vec{x}^{(e)})$  :

## Parametric

model dependent:  $\hat{f}(\vec{x}; \{\alpha\})$   
=> determine the parameters  
 $\{\alpha\}$  with respect to some  
goodness-of-fit criterion ( $\chi^2$ ,  
likelihood,...)

## Non-Parametric

goal: remove model dependence  
e.g. histogram, Kernel PDE,...

## Some popular PDE methods...

Histogram: the simplest method. Poor accuracy unless the function varies smoothly within each bin. Difficult to use in n-D:  $N_{\text{data}} \sim N_{\text{bin}}^n$

Fixed Kernel PDE:  $\hat{f}_0(x) = \frac{1}{nh} \sum_{e=1}^N K\left(\frac{x - x^{(e)}}{h}\right)$ ;  $h = \text{bandwidth}$

- straightforward extension to n-D.
- if  $K = 1(0)$  if  $x \in (\epsilon)$  hypercube => importance sampling histogramming ( $h = \text{side of the hypercube}$ )
- very popular to use  $K = G(0,1)$  => remove discontinuities ( $h$  spreads out the contribution from each data point)

Adaptive Kernel PDE:

- to make  $h$  depend on the data locally
- "second iteration" to the general Kernel PDE method.  
=> in general the Kernel method has become very popular:
  - in general unbinned
  - depends only on the data sample used (~model independent, but some user freedom exists: choice of  $h$ , Kernel function)
  - rather accurate (BUT problematic when hard boundaries are present)
  - in general rather slow



## What about parametric ones?

- In general:
  - simple-minded: e.g. Multi-Gaussian-Expansion (determine number of classes and gaussian parameters from a likelihood fit to the data sample).
  - model-dependent with increasing difficulty to "guess" the model as the dimensionality increases.
  - not very accurate.
  - BUT fast evaluation of the estimated pdf.
- QUESTION: is there a PDE method with the potential of Kernel PDE and allowing for a fast evaluation of the estimated pdf?
- Idea => make use of NNs (NNPDE) to estimate pdfs:
  - in n-dimensions,
  - from examples,
  - in an unbinned way,
  - with an analytical expression,
  - ~model-independent,
  - with the possibility to quantify a "goodness-of-fit".

Reference: Ll. Garrido and A. Juste, *Comput. Phys. Commun.* 115 (1998)

## NNPDE: Method

- The method **takes advantage of the statistical interpretation of the NN output in classification problems.**
- Consider a sample of  $N$  events distributed among 2 classes ( $C_1$  and  $C_2$ ), each event  $e$  being characterized by a set of  $n$  variables  $x^{(e)}$ . Each class of patterns has a proportion  $\alpha_i$  and is generated by the normalized pdf  $P_i(x)$  ( $= P(x | C_i)$ ).
- It is well known that by minimizing over this sample the quadratic output error:

$$E[\rho] = \frac{1}{2N} \sum_{e=1}^N [\rho(x^{(e)}) - d(x^{(e)})]^2$$

$$d(x) = 1 \text{ for } x \in C_1; 0 \text{ for } x \in C_2$$

with respect to the **unconstrained function  $\rho(x)$ , the minimum is achieved when:**

$$\rho^{(\min)}(x) = P(C_1 | x)$$

- This procedure is usually done by using **layered feed-forward NNs**.
- Here I consider NNs with topologies:  $n - N_{h1} - N_{h2} - 1$

## NNPDE: Method

- Let us consider we have a large amount of events ("data") distributed according to the unknown pdf  $P_{data}(x)$ , whose analytical expression is unknown and which we want precisely to estimate.
- If a NN is trained to disentangle between the sample of "data" events and another sample ("reference") generated according to any known pdf  $P_{ref}(x)$ , the NN output will approximate, after training,

$$o_{NN}^{(\min)}(x) = \hat{P}(data \mid x) = \frac{\alpha_{data} \hat{P}_{data}(x)}{\alpha_{data} \hat{P}_{data}(x) + \alpha_{ref} \hat{P}_{ref}(x)}; \quad \alpha_{data} + \alpha_{ref} = 1$$

where  $\alpha_{data}$  and  $\alpha_{ref}$  are the proportions used for training.

- Since  $P_{ref}(x)$  is known, the NN approximation to  $P_{data}(x)$  is given by:

$$\hat{P}_{data}(x) = P_{ref}(x) \frac{\alpha_{ref}}{\alpha_{data}} \frac{o_{NN}^{(\min)}(x)}{1 - o_{NN}^{(\min)}(x)}$$

CONDITION :  $P_{ref}(x) \neq 0 \quad \forall x / \quad P_{data}(x) \neq 0$

## NNPDE: Method

- As a result,  $P_{\text{data}}(x)$ :
  - is determined in an **unbinned** way from examples.
  - has an **analytical** expression (indeed we have it for  $P_{\text{ref}}(x)$  and  $\sigma_{\text{NN}}(x)$ )  
=> **fast computation!!** (in contrast to Kernel estimation methods)
  - in principle **properly normalized**. The normalization of  $P_{\text{data}}(x)$  depends on the goodness of the NN approximation to the a-posteriori Bayesian probability. Since  $\int dx P_{\text{ref}}(x) = 1 \Rightarrow \int dx P_{\text{data}}(x) \approx 1$
- What about  $P_{\text{ref}}(x)$ ?
  - A priori it can be any pdf, e.g. a flat distribution,  
BUT due to limitations in the approximation of the NN to the a-posteriori Bayesian probability (finite available statistics for training, limited flexibility from network architecture, minimization algorithm, etc)  
=> Use a pdf built from the product of normalized good approximations to each 1-D projection of the data pdf.  
=> Makes easier the learning of complex correlations in the n-D space.

## NNPDE: Goodness of Fit

- Given a data sample containing  $N_{\text{data}}$  events, it is possible to perform a test of the hypothesis of the data sample under consideration being consistent with coming from the mapped pdf;
- Generate MC samples according to the mapped pdf and containing  $N_{\text{data}}$  events.
- Compute the distribution of the log-likelihood function from the MC samples

$$L = \log(L) = \sum_{e=1}^{N_{\text{data}}} \log(\hat{P}_{\text{data}}(x^{(e)})) \Rightarrow P(L)$$

- Being  $L_{\text{data}}$  the value of the log-likelihood for the original data sample, the confidence level associated to the hypothesis of the data sample coming from the mapped pdf is given by:

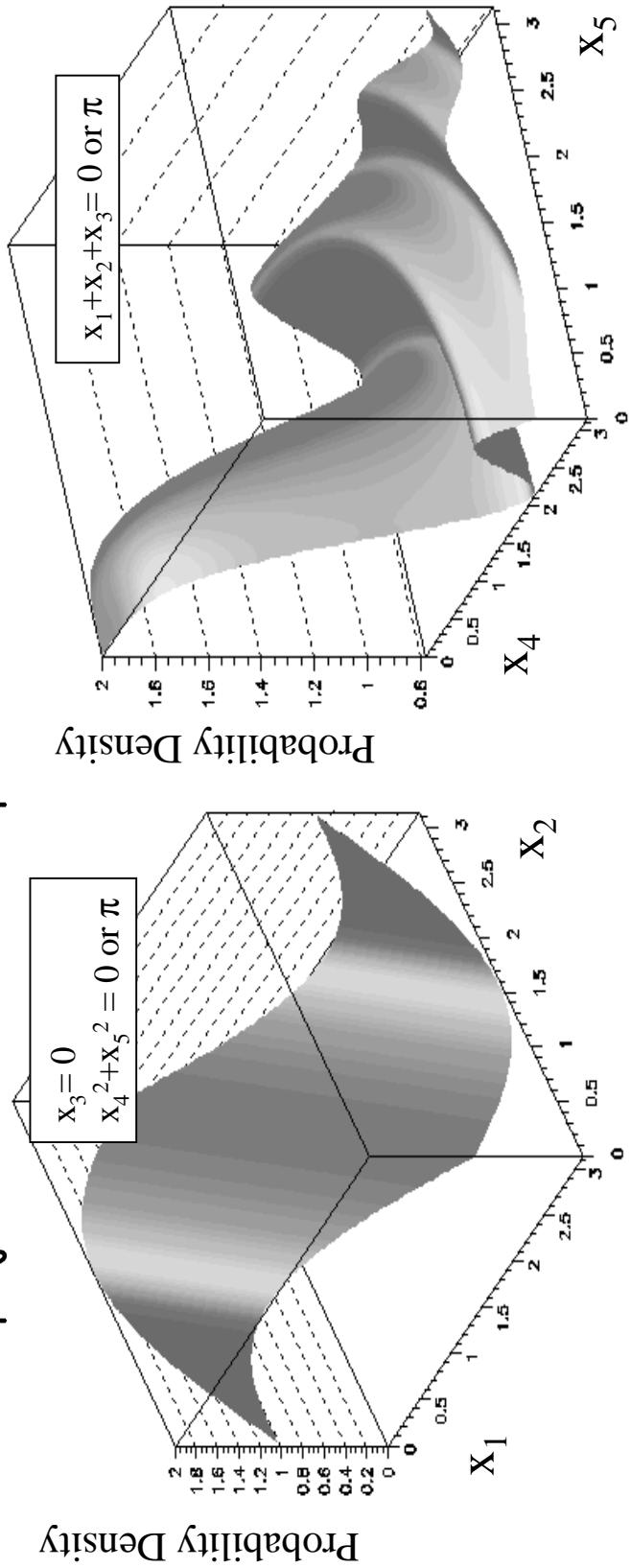
$$CL = \int_{L_{\text{data}}}^{\infty} dL P(L)$$

=> If the mapped pdf is a good approximation to  $P_{\text{data}}(x)$ , the expected distribution for CL evaluated for different data samples should be flat (as it corresponds to a cumulative probability).

## NNPDE: Artificial Example

- Want to map the following 5-D pdf from examples:
$$P_{data}(\vec{x}) = \frac{1}{C} \left[ \sin(x_1 + x_2 + x_3) + 1 \right] \left[ \frac{\sin(x_4^2 + x_5^2)}{x_4^2 + x_5^2} + 1 \right]$$

$$\vec{x} = (x_1, x_2, x_3, x_4, x_5) \in [0, \pi]^5 \subset \Re^5$$
- Rather intricate structure of peaks and valleys in the 5-D space.
- Large variations in the "local covariance matrix" from point to point.
- Some projections into a 2-D subspace:



## NNPDE: Artificial Example

- Generate 100k events distributed according to  $P_{\text{data}}(x)$ .
- Choose a suitable  $P_{\text{ref}}(x)$ : the 1-D projections of the first 3 variables are equal and essentially flat, whereas the 1-D projections of the last 2 variables can be parameterized as a 4<sup>th</sup> polynomial

$$P_{\text{ref}}(\vec{x}) = \frac{1}{C} p_4(x_4) p_4(x_5)$$

and generate 100k events.

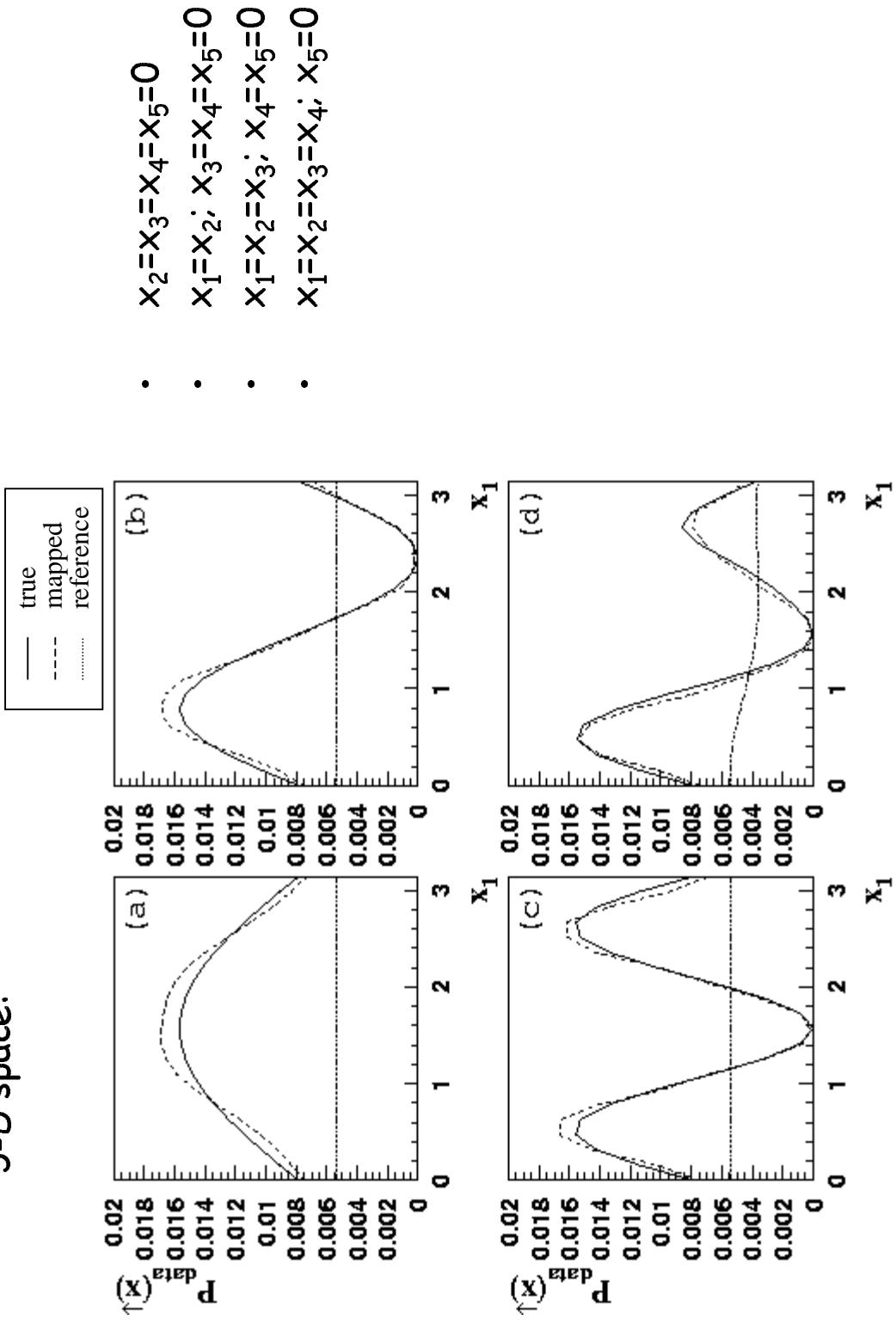
- Train a NN to disentangle between the "data" and the "reference" samples.
- After training, the estimated  $P_{\text{data}}(x)$  is given by ( $\alpha_{\text{data}} = \alpha_{\text{ref}} = 1/2$ ):

$$\hat{P}_{\text{data}}(\vec{x}) = P_{\text{ref}}(\vec{x}) \frac{o_{NN}^{(\min)}(\vec{x})}{1 - o_{NN}^{(\min)}(\vec{x})}$$

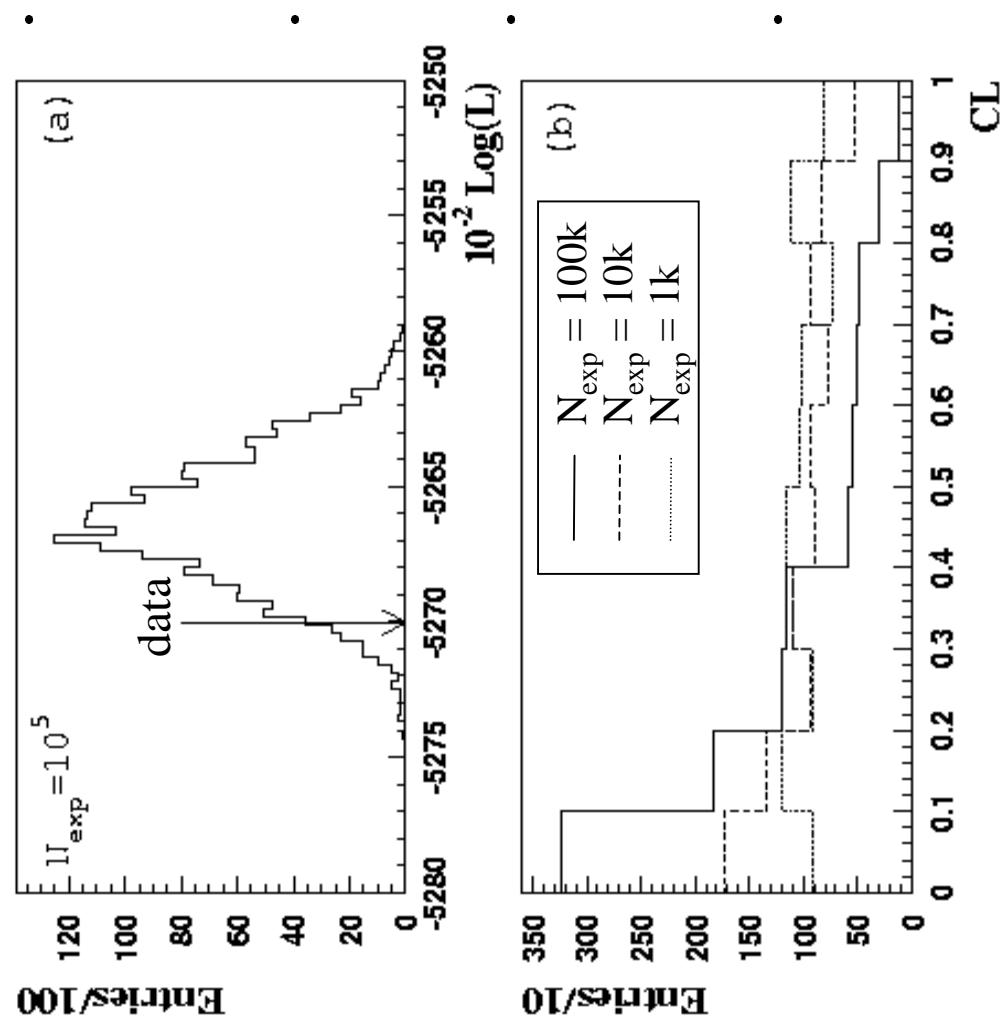
- The normalization of the estimated pdf is consistent with 1 at the 1% level.

## NNPDE: Artificial Example

- Comparison between the true and mapped pdfs in different slices of the 5-D space:

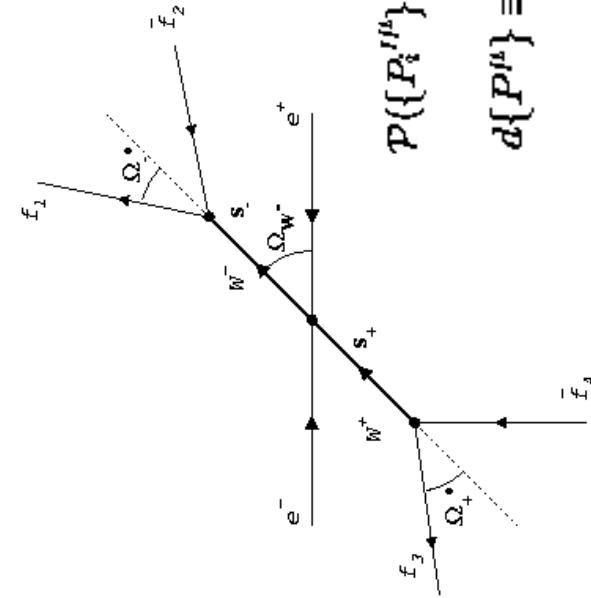


## NNPDE: Artificial Example



- The CL of the original data sample (100k events) of being consistent with coming from the mapped pdf is 5.5% (kind of low, an statistical fluctuation?)
- Since the true pdf is known, the distribution of CL assuming different data sample sizes can be obtained.
- Indeed, data samples of a size comparable to the one used for training have enough resolution to detect systematic deviations in the mapped pdf.
- In HEP, one typically estimates pdfs from MC samples  $\sim 100$  times larger than the experimental data sample at hand. In that case the mapped pdf would be perfectly suitable.

## NNPDE: HEP Example



- W mass determination through direct reconstruction at LEP2.
- Consider  $e^+e^- \rightarrow W^+W^- \rightarrow jjjj$  events and determine  $M_W$  from a maximum likelihood fit to the whole event topology (matrix-element approach).

$$\mathcal{P}(\{P_i'^\mu\} \mid M_W) = \int \cdots \int d\{P_i'^\mu\} T(\{P_i'^\mu\} \mid \{P_i^\mu\}; M_W) \mathcal{P}(\{P_i^\mu\} \mid M_W),$$

$$d\{P^\mu\} \equiv ds_1 ds_2 d\Omega_W d\Omega_1^* d\Omega_3^* dx$$

$$\frac{d^9 \sigma}{ds_+ ds_- d\Omega_W - d\Omega_-^* d\Omega_+^* dx} (\{P_i'^\mu\} \mid M_W) = \frac{(2\pi)^4 \lambda^{1/2}(s', s_+, s_-)}{2s' \cdot 512s'(2\pi)^{12}} \mid \mathcal{M}_{CC03} \mid^2 (\{P_i'^\mu\}; M_W).$$

$$\cdot \quad \mathcal{H}(x, s)(1 + \delta_C(s', s_+, s_-; M_W)).$$

- The event probability takes into account physics and combinatorial backgrounds effects as well as resolution effects (transfer function).
- It is for the determination of the transfer function that PDE through NNs becomes useful (in particular need a fast evaluation of the conditional probability!!!).

## NNPDE: HEP Example

- In this example the transfer function can be simplified:

$$T(\{P_i'^{\mu}\} | \{P^{\mu}\}; M_W) \equiv T(s_1', s_2', \Omega_W', \Omega_1'^*, \Omega_3'^* | s_1, s_2, \Omega_W, \Omega_1^*, \Omega_3^*)$$



$$T(s_1', s_2', \Omega_W', \Omega_1'^*, \Omega_3'^* | s_1, s_2, \Omega_W, \Omega_1^*, \Omega_3^*) \simeq T_S(s_1', s_2' | s_1, s_2, x) \cdot \delta(\Omega_W' - \Omega_W) \\ \cdot \delta(\Omega_1'^* - \Omega_1^*) \delta(\Omega_3'^* - \Omega_3^*).$$

and the problem is to determine the transfer function for the invariant masses.

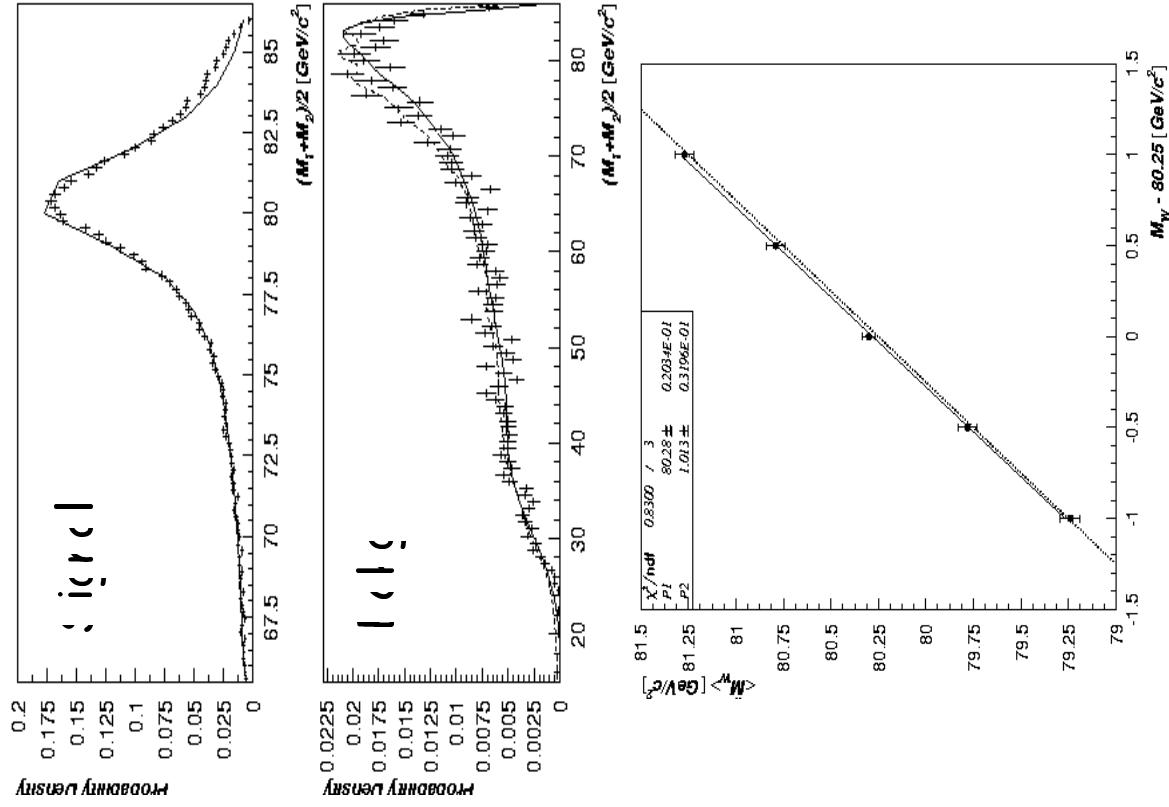
$$T_S(s_1', s_2' | s_1, s_2, x) = \frac{f_S(s_1', s_2', s_1, s_2, x | M_W^0)}{g_S(s_1, s_2, x | M_W^0)}$$

Map with NNs

Theoretical expression known

- Generate 40k hadronic WW fully reconstructed events ("data") and train a NN to disentangle them from 40k events generated according to the product of 1-D projections of the MC sample ("reference").
- Compare the final predicted distribution (after folding of resolution effects) for some 1-D projections (7-D integrals required) with the MC expectation. In this particular example, observe some problems trying to reproduce the large correlations (at the high boundary) in the 2-D space of the reconstructed W invariant masses.

# NNPDF: HEP Example



- The inclusive background pdf is mapped out in similar way.
- The normalization of the mapped pdfs is consistent with 1 at the 3% level.
- Some small systematic differences observed between the MC expectation and the prediction in the 1-D projections in regions which shouldn't have a large impact on the estimated  $M_W$ .
- In the multidimensional space where the fit is performed (8-D), the sensitivity to small systematic effects is washed out => advantage of increasing the dimensionality of the fitting space.
- In addition, the data sample to be fitted (70 events) is small enough as not to have "enough resolution" to detect those discrepancies => resulting likelihood estimator is unbiased.

## Some Remarks

- The examples shown before were done using the standard back-propagation algorithm (learning method) and simple gradient descent (minimization of the error function) => the application of recent developments in minimization algorithms (conjugate gradients, BFGS, hybrid, etc) will help the NN to converge better:  
=> better approximation to Bayes probability => better approximation for mapped pdf to the true one
- The interpolating nature of the NN approximation produces continuous pdf estimates and allows to fully exploit the finite statistics available for training.
- The analytical expression obtained for the mapped pdf allows fast and efficient computation, essential for its application in complicated fitting formulas (multidimensional folding of resolution effects, etc).
- The method doesn't underestimate the pdf close to hard boundaries (as for Kernel PDE) since  $P_{ref}(x)$  is by construction =0 outside the domain of the parent distribution. Therefore, no tricks need to be applied in order to get a properly normalized pdf.
- The freedom in choosing the NN architecture is comparable to the freedom in Kernel PDE methods in choosing the kernel and the bandwidth parameter.

## Conclusions

- NNs offer another avenue to the problem of multidimensional pdf estimation.
- The method is simple and exploits the NN interpretation in terms of a-posteriori Bayesian probability when an unary representation is taken for the output patterns.
- As a result, the mapped pdf:
  - is determined from examples in an unbinned way;
  - is determined with no a-priori knowledge on the form of the parent distribution required ( $P_{\text{ref}}(x)$  can just be a flat distribution);
  - has an analytical expression, which results in a fast and efficient computation;
  - is properly normalized;
  - can estimate the goodness of fit of the mapped pdf to the "generating data sample". This depends on the appropriate NN convergence and the use of more sophisticated minimization algorithms will only help.
- Currently studying a method to train a NN whose output will give directly a conditional probability:  $P(\vec{x} \mid \vec{y}) \Rightarrow$  e.g. direct mapping of "transfer functions"